

LAPORAN AKHIR

PROYEK



DML (DATA MANIPULATION LANGUAGE) DAN JOIN TABLE

Disusun Oleh:

Nama : 11117320

NPM : Chantika Amanda

Kelas :

LEMBAGA PENGEMBANGAN KOMPUTERISASI
UNIVERSITAS GUNADARMA

2022

DAFTAR ISI

DESKRIPSI LAPORAN	3
TAHAPAN Pengerjaan.....	4
PENUTUP	17



DESKRIPSI LAPORAN

DML (DATA MANIPULATION LANGUAGE) DAN JOIN TABLE

Pada pertemuan 2 laporan berisikan mengenai cara melakukan DML seperti SELECT, INSERT, UPDATE, DELETE selain itu seperti cara membuat Table, Tipe Data, Database Diagram, Constraint yang ada pada Table, Join Table



TAHAPAN Pengerjaan

1. Ringkasan Materi

DML (Data Manipulation Language) digunakan untuk memanipulasi data yang ada dalam suatu table:

- SELECT untuk menampilkan data
- INSERT untuk menambahkan data baru
- UPDATE untuk mengubah data yang sudah ada
- DELETE untuk menghapus data

SQL Server menyediakan banyak cara untuk bekerja dengan tabel. Membuat tabel baru dengan menggunakan fitur New Table di SQL Server Management Studio menggunakan perintah CREATE TABLE, memodifikasi tabel yang ada menggunakan fitur Modify Table di SQL Server Management Studio dengan perintah ALTER TABLE, serta melakukan fungsi manajemen table lainnya, termasuk menyalin, mengganti nama, dan menghapus tabel.

Sebelum membuat tabel, perlu diperhatikan dalam penggunaan nama tabel. Nama tabel dapat memuat sampai 128 karakter. Nama tabel harus dimulai dengan karakter alfabet dan mengandung underscore (_), simbol (@), tanda pagar (#) dan nomor. Pengecualian di dalam aturan ini terdapat pada tabel temporary. Tabel temporary local, memiliki nama yang dimulai dengan # dan di akses hanya ke beberapa sesi pengguna. Tabel temporary global memiliki nama yang dimulai dengan ## dan di akses kepada siapa pun selama sesi pengguna tetap terhubung. Tabel temporary dibuat di dalam database tempdb dan dihapus secara otomatis saat sesi pengguna berakhir. Nama tabel harus unik untuk setiap skema di dalam sebuah database. Namun skema dapat berisi like-named tables. Sehingga dapat membuat tabel multiple contacts selama file tersebut didefinisikan dalam skema terpisah. Contohnya, Customer, Employee, dan Servis yang dimiliki oleh tabel contacts

Beberapa Constraint yang sering digunakan

- NOT NULL Menentukan bahwa kolom tidak dapat menerima nilai null. Namun, nilai null bukan nilai 0, tetapi nilai null adalah yang tidak diisi atau tidak terdefinisi.
- Primary Key Satu atau lebih kolom pada tabel yang memiliki baris yang eksklusif di dalam tabel akan membentuk primary key. Spesifikasi dari primary key memastikan integritas dari tabel. Kolom yang membentuk primary key tidak bisa mengandung nilai null. Apabila menentukan nilai

sebagai primary, SQL Server akan membuat indeks tertentu untuk memastikan bahwa kunci tersebut bersifat eksklusif. Tabel yang merupakan primary key ditandai dengan adanya tanda kunci disamping kolom.

- Foreign Key Menentukan hubungan antara tabel-tabel. Foreign Key adalah field di table aktif yang menunjukan ke field kunci pada tabel lain. Penggunaan foreign key mampu mencegah penghapusan baris-baris pada tabel yang aktif, apabila ada referensi field-field kunci dari tabel eksternal. Batasan semacam ini mencegah nilai yang dimasukkan ke dalam tabel yang tidak ditentukan pada field kunci dari tabel eksternal yang digabungkan.

Diagram database adalah representasi dari sebuah database. Dengan diagram database bisa mengelola database dengan menggunakan antarmuka grafis serta menggeser mouse untuk melakukan tugas-tugas yang biasanya dilakukan oleh perintah bahasa SQL. Tugas-tugas ini meliputi menambah atau menghapus tabel, trigger prosedur, batasan kolom tabel, serta hubungan antar tabel.

JOIN merupakan salah satu operasi untuk menghubungkan dua atau lebih tabel yang saling berelasi. JOIN sendiri memiliki berbagai jenis, diantaranya : JOIN, INNER JOIN, LEFT JOIN, LEFT OUTER JOIN, RIGHT JOIN, RIGHT OUTER JOIN, CROSS JOIN, dan JOIN dengan koma.

- JOIN / INNER JOIN merupakan jenis JOIN yang digunakan untuk mendapatkan data dari dua tabel atau lebih yang persis saling berelasi. Hasil dari JOIN dengan menggunakan JOIN / INNER JOIN merupakan irisan dari relasi antar tabel.
- LEFT JOIN / LEFT OUTER JOIN merupakan jenis JOIN yang digunakan untuk mendapatkan data dari dua tabel atau lebih dimana data di tabel sebelah kiri ditampilkan semua baik yang berelasi dengan data di tabel sebelah kanan maupun tidak
- RIGHT JOIN / RIGHT OUTER JOIN merupakan jenis JOIN yang digunakan untuk mendapatkan data dari dua tabel atau lebih dimana data di tabel sebelah kanan ditampilkan semua baik yang berelasi dengan data di tabel sebelah kiri maupun tidak
- CROSS JOIN merupakan jenis JOIN yang digunakan untuk mendapatkan data kombinasi dari dua tabel atau lebih. Misalkan, n = jumlah baris data pada tabel di sebelah kiri, dan m = jumlah baris data pada tabel di sebelah kanan. Maka hasil jumlah baris dari CROSS JOIN adalah $n \times m$ baris data.
- Selain menggunakan operator JOIN, dapat juga melakukan JOIN antar tabel dengan menggunakan koma. Bila tidak menambahkan kondisi relasi dengan menggunakan WHERE, maka JOIN dengan koma akan menghasilkan data yang sama apabila menggunakan CROSS JOIN

2. Langkah-Langkah

1. Buat Table (Script Terlampir pada Zip File)

```

5
6
7 -- No 1
8
9 CREATE TABLE regions
10 ( region_id      NUMERIC(3) PRIMARY KEY
11   CONSTRAINT region_id_nn NOT NULL
12   , region_name   VARCHAR(25)
13 );
14
15 CREATE TABLE countries
16 ( country_id      CHAR(2)
17   CONSTRAINT country_id_nn NOT NULL
18   , country_name   VARCHAR(40)
19   , region_id      NUMERIC
20   CONSTRAINT country_c_id_pk
21                 PRIMARY KEY (country_id)
22 );
23
24 CREATE TABLE locations
25 ( location_id     NUMERIC(4)
26   CONSTRAINT location_id_nn NOT NULL
27   , street_address VARCHAR(40)
28   , postal_code    VARCHAR(12)
29   , city           VARCHAR(30)

```

2. Masukkan Data

```

87
88 -- No 2
89
90 INSERT INTO regions VALUES
91 ( 1
92   , 'Europe'
93 );
94
95 INSERT INTO regions VALUES
96 ( 2
97   , 'Americas'
98 );
99
100 INSERT INTO regions VALUES
101 ( 3
102   , 'Asia'
103 );
104
105 INSERT INTO regions VALUES
106 ( 4
107   , 'Middle East and Africa'
108 );
109 select * from regions;
110
111 INSERT INTO countries VALUES
112 ( 'IT'
113   , 'Italy'

```

3. Tampilkan data karyawan yang memiliki nama berawalan huruf "M". Dengan menggunakan Clause WHERE LIKE maka kita dapat menampilkan Karyawan yang berawalan dengan M

```
2389  
2390 SELECT * FROM employees WHERE first_name LIKE 'M%'  
2391  
2392
```

	employee_id	first_name	last_name	email	phone_NUMERIC	hire_date	job_id	salary	commission_pct	manager_id	department_id
1	120	Matthew	Weiss	MWEISS	650.123.1234	1996-07-18	ST_MAN	8000.00	NULL	100	50
2	130	Mozhe	Atkinson	MATKINSO	650.124.6234	1997-10-30	ST_CLERK	2800.00	NULL	121	50
3	134	Michael	Rogers	MROGERS	650.127.1834	1998-08-26	ST_CLERK	2900.00	NULL	122	50
4	164	Mattea	Marvins	MMARVINS	011.44.1346.329268	2000-01-24	SA_REP	7200.00	0.10	147	80
5	182	Martha	Sullivan	MSULLIVA	650.507.9878	1999-06-21	SH_CLERK	2500.00	NULL	120	50
6	201	Michael	Hartstein	MHARTSTE	515.123.5555	1996-02-17	MK_MAN	13000.00	NULL	100	20

4. Tampilkan data karyawan yang mulai bekerja antara tanggal 20 Mei 1991 dan 17 Juni 1995. Dengan menggunakan Clause WHERE dan BETWEEN kita dapat mencari suatu nilai diantara 2 nilai lain

```
2392  
2393 SELECT * FROM employees WHERE hire_date BETWEEN '1991/05/20' AND '1995/06/17'
```

	employee_id	first_name	last_name	email	phone_NUMERIC	hire_date	job_id	salary	commission_pct	manager_id	department_id
1	102	Lex	De Haan	LDEHAAN	515.123.4569	1993-01-13	AD_VP	17000.00	NULL	100	90
2	104	Bruce	Ernst	BERNST	590.423.4568	1991-05-21	IT_PROG	6000.00	NULL	103	60
3	108	Nancy	Greenberg	NGREENBE	515.124.4569	1994-08-17	FI_MGR	12000.00	NULL	101	100
4	109	Daniel	Faviet	DFAVIET	515.124.4169	1994-08-16	FI_ACCOUNT	9000.00	NULL	108	100
5	114	Den	Raphaely	DRAPHEAL	515.127.4561	1994-12-07	PU_MAN	11000.00	NULL	100	30
6	115	Alexander	Khoo	AKHOO	515.127.4562	1995-05-18	PU_CLERK	3100.00	NULL	114	30
7	122	Payam	Kaufling	PKAUFLIN	650.123.3234	1995-05-01	ST_MAN	7900.00	NULL	100	50
8	203	Susan	Mavris	SMAVRIS	515.123.7777	1994-06-07	HR_REP	6500.00	NULL	101	40
9	204	Hermann	Baer	HBAER	515.123.8888	1994-06-07	PR_REP	10000.00	NULL	101	70
10	205	Shelley	Higgins	SHIGGINS	515.123.8080	1994-06-07	AC_MGR	12000.00	NULL	101	110
11	206	William	Gietz	WGIEZT	515.123.8181	1994-06-07	AC_ACCOU...	8300.00	NULL	205	110

5. Ganti tipe data region_id menjadi numeric(3) pada tabel countries, kemudian jadikan sebagai foreign key. Alter Pertama untuk mengubah tipe data menjadi Numeric(3) lalu jalankan Alter kedua yang menambahkan Constraint Foreign Key

```
2402
2403 -- No 5
2404
2405 ALTER TABLE countries
2406 ALTER COLUMN region_id NUMERIC(3)
2407 GO
2408 ALTER TABLE countries
2409 ADD CONSTRAINT countr_reg_fk FOREIGN KEY (region_id) REFERENCES regions(region_id)
2410
2411
2412 -- No 6
```

Messages
Command(s) completed successfully.

6. Gunakan inner join untuk menampilkan region_name dan country_name pada tabel regions dan countries. Inner Join dengan Syarat region_id untuk menggabungkan data 2 table

```
2416
2417 -- No 6
2418
2419 SELECT region_name, country_name FROM regions A JOIN countries B ON A.region_id = b.region_id
```

Results

	region_name	country_name
1	Americas	Argentina
2	Asia	Australia
3	Europe	Belgium
4	Americas	Brazil
5	Americas	Canada
6	Europe	Switzerland
7	Asia	China
8	Europe	Germany
9	Europe	Denmark
10	Middle East and Africa	Egypt
11	Europe	France
12	Asia	HongKong
13	Middle East and Africa	Israel
14	Asia	India
15	Europe	Italy
16	Asia	Japan
17	Middle East and Africa	Kuwait
18	Americas	Mexico

7. Buat primary key pada location_id di tabel locations serta foreign key di country_id. Alter table untuk menambahkan Constraint Primary Key lalu dilanjutkan penambahan Foreign Key


```
2420  
2421 ALTER TABLE locations  
2422 ADD CONSTRAINT loc_id_pk PRIMARY KEY (location_id),  
2423 CONSTRAINT loc_c_id_fk FOREIGN KEY (country_id) REFERENCES countries(country_id);
```

Messages

Command(s) completed successfully.

8. Tampilkan city dan street_address pada tabel locations, serta country_name pada tabel countries dengan menggunakan left outer join urutkan secara ascending berdasarkan city pada tabel locations. Dengan Menggunakan LEFT OUTER JOIN untuk menampilkan seluruh data table kiri dan data bersinggungan dengan table kanan dan lalu dengan ORDER BY kita dapat mengurutkan urutan data yang muncul berdasarkan kolom



```

2426
2427 SELECT city, street_address, country_name
2428 FROM locations A LEFT OUTER JOIN countries B ON A.country_id = B.country_id
2429 ORDER BY A.city ASC

```

Results Messages			
	city	street_address	country_name
1	Beijing	40-5-12 Laogianggen	China
2	Bern	Murtenstrasse 921	Switzerland
3	Bombay	1298 Vileparle (E)	India
4	Geneva	20 Rue des Corps-Saints	Switzerland
5	Hiroshima	9450 Kamiya-cho	Japan
6	London	8204 Arthur St	United Kingdom
7	Mexico City	Mariano Escobedo 9991	Mexico
8	Munich	Schwanthalerstr. 7031	Germany
9	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom
10	Roma	1297 Via Cola di Rie	Italy
11	Sao Paulo	Rua Frei Caneca 1360	Brazil
12	Seattle	2004 Charade Rd	United States of America
13	Singapore	198 Clementi North	Singapore
14	South Brunswick	2007 Zagora St	United States of America
15	South San Francisco	2011 Interiors Blvd	United States of America
16	Southlake	2014 Jabberwocky Rd	United States of America
17	Stretford	9702 Chester Road	United Kingdom
18	Sydney	12-98 Victoria Street	Australia

9. Pada tabel departments, ganti tipe data locations_id menjadi NUMERIC(4). Kemudian tambahkan primar key pada department_id dan foegin key pada loction_id pada tabel departments. Alter Pertama untuk mengubah tipe Data lalu dilanjutkan dengan penambahan Constraint Primary Key dan Foreign Key

```
2430
2431 -- No 9
2432
2433 ALTER TABLE departments
2434 ALTER COLUMN location_id NUMERIC(4)
2435 GO
2436 ALTER TABLE departments
2437 ADD CONSTRAINT dept_id_pk PRIMARY KEY (department_id),
2438 CONSTRAINT dept_loc_fk FOREIGN KEY (location_id) REFERENCES locations(location_id)
```

Messages

Command(s) completed successfully.

10. Tampilkan street_address dan postal_code pada tabel locations serta department_name pada tabel departments, tapi tampilkan postal_code yang memiliki no 98199 saja. Dengan menggunakan JOIN untuk menggabungkan data dan menggunakan WHERE untuk filter postal_code



2414	-- No 10
2415	
2416	SELECT street_address, postal_code, department_name
2417	FROM locations A JOIN departments B ON A.location_id = B.location_id
2418	WHERE A.postal_code = '98199'

Results		Messages	
	street_address	postal_code	department_name
1	2004 Charade Rd	98199	Administration
2	2004 Charade Rd	98199	Purchasing
3	2004 Charade Rd	98199	Executive
4	2004 Charade Rd	98199	Finance
5	2004 Charade Rd	98199	Accounting
6	2004 Charade Rd	98199	Treasury
7	2004 Charade Rd	98199	Corporate Tax
8	2004 Charade Rd	98199	Control And Credit
9	2004 Charade Rd	98199	Shareholder Services
10	2004 Charade Rd	98199	Benefits
11	2004 Charade Rd	98199	Manufacturing
12	2004 Charade Rd	98199	Construction
13	2004 Charade Rd	98199	Contracting
14	2004 Charade Rd	98199	Operations
15	2004 Charade Rd	98199	IT Support
16	2004 Charade Rd	98199	NOC
17	2004 Charade Rd	98199	IT Helpdesk
18	2004 Charade Rd	98199	Government Sales
19	2004 Charade Rd	98199	Retail Sales

11. Tampilkan job title, min salary, max salary pada tabel jobs, Serta first name dan salary pada tabel employee, dengan salary lebih kecil sama dengan 6000. Dengan menggunakan JOIN untuk menggabungkan data dan WHERE untuk filter salary

```

2420 -- No 11
2421
2422 SELECT job_title, min_salary, max_salary, first_name, salary
2423 FROM jobs A JOIN employees B ON A.job_id = B.job_id
2424 WHERE salary <= 6000

```

Results		Messages			
	job_title	min_salary	max_salary	first_name	salary
1	Programmer	4000	10000	Bruce	6000.00
2	Programmer	4000	10000	David	4800.00
3	Programmer	4000	10000	Valli	4800.00
4	Programmer	4000	10000	Diana	4200.00
5	Purchasing Clerk	2500	5500	Alexander	3100.00
6	Purchasing Clerk	2500	5500	Shelli	2900.00
7	Purchasing Clerk	2500	5500	Sigal	2800.00
8	Purchasing Clerk	2500	5500	Guy	2600.00
9	Purchasing Clerk	2500	5500	Karen	2500.00
10	Stock Manager	5500	8500	Kevin	5800.00
11	Stock Clerk	2000	5000	Julia	3200.00
12	Stock Clerk	2000	5000	Irene	2700.00
13	Stock Clerk	2000	5000	James	2400.00
14	Stock Clerk	2000	5000	Steven	2200.00
15	Stock Clerk	2000	5000	Laura	3300.00
16	Stock Clerk	2000	5000	Mozhe	2800.00
17	Stock Clerk	2000	5000	James	2500.00
18	Stock Clerk	2000	5000	TJ	2100.00
19	Stock Clerk	2000	5000	Jason	3300.00
20	Stock Clerk	2000	5000	Michael	2900.00
21	Stock Clerk	2000	5000	Ki	2400.00
22	Stock Clerk	2000	5000	Hazel	2200.00
23	Stock Clerk	2000	5000	Renske	3600.00

12. Tampilkan nama belakang, gaji bulanan pegawai dan kode departemen tempat pegawai tersebut bekerja khusus untuk pegawai yang bekerja pada departemen dengan kode 100 atau 101, dan memiliki gaji lebih besar atau sama dengan 1000. Dengan menggunakan JOIN untuk menggabungkan data dan menggunakan WHERE IN IN dapat digunakan untuk kondisi yang banyak

```
2426 -- No 12
2427
2428 SELECT last_name, salary, C.department_id
2429 FROM employees A JOIN jobs B ON A.job_id = B.job_id
2430 JOIN departments C ON C.department_id = A.department_id
2431 WHERE c.department_id IN (100, 101) AND salary >= 1000
```

<

Results Messages

	last_name	salary	department_id
1	Greenberg	12000.00	100
2	Faviet	9000.00	100
3	Chen	8200.00	100
4	Sciarra	7700.00	100
5	Uman	7800.00	100
6	Popp	6900.00	100

13. Tampilkan job title, min salary pada tabel jobs dan first name, salary pada tabel employees, serta department name pada tabel departments yang memiliki first name berawalan huruf "S". Dengan menggunakan JOIN untuk menggabungkan data dan juga menggunakan WHERE LIKE untuk mendapatkan data employee dengan nama depan S

```

2433 -- No 13
2434
2435 SELECT job_title, min_salary, first_name, salary, department_name
2436 FROM jobs A JOIN employees B ON A.job_id = B.job_id
2437 JOIN departments C ON C.department_id = B.department_id
2438 WHERE first_name LIKE 'S%'

```

Results		Messages			
	job_title	min_salary	first_name	salary	department_name
1	President	20000	Steven	24000.00	Executive
2	Accounting Manager	8200	Shelley	12000.00	Accounting
3	Sales Representative	6000	Sarath	7000.00	Sales
4	Sales Representative	6000	Sundar	6400.00	Sales
5	Sales Representative	6000	Sundita	6100.00	Sales
6	Purchasing Clerk	2500	Shelli	2900.00	Purchasing
7	Purchasing Clerk	2500	Sigal	2800.00	Purchasing
8	Stock Manager	5500	Shanta	6500.00	Shipping
9	Stock Clerk	2000	Steven	2200.00	Shipping
10	Stock Clerk	2000	Stephen	3200.00	Shipping
11	Shipping Clerk	2500	Sarah	4000.00	Shipping
12	Shipping Clerk	2500	Samuel	3200.00	Shipping
13	Human Resources Representative	4000	Susan	6500.00	Human Resources

14. Tampilkan last name, hire date, commission pct pada tabel employee, department name pada tabel department, serta city, state provincy dan country name pada tabel countries. Tampilkan commission pct NOT NULL. Dengan menggunakan JOIN dan WHERE IS NOT NULL disini kita dapat filter kolom yang valuenya tidak NULL

```

2440 -- No 14
2441
2442 SELECT last_name, hire_date, commission_pct, department_name, city, state_province, country_name
2443 FROM employees A JOIN departments B ON A.department_id = B.department_id
2444 JOIN locations C ON C.location_id = B.location_id
2445 JOIN countries D ON D.country_id = C.country_id
2446 WHERE commission_pct IS NOT NULL

```

	last_name	hire_date	commission_pct	department_name	city	state_province	country_name
1	Russell	1996-10-01	0.40	Sales	Oxford	Oxford	United Kingdom
2	Partners	1997-01-05	0.30	Sales	Oxford	Oxford	United Kingdom
3	Errazuriz	1997-03-10	0.30	Sales	Oxford	Oxford	United Kingdom
4	Cambrault	1999-10-15	0.30	Sales	Oxford	Oxford	United Kingdom
5	Zlotkey	2000-01-29	0.20	Sales	Oxford	Oxford	United Kingdom
6	Tucker	1997-01-30	0.30	Sales	Oxford	Oxford	United Kingdom
7	Bernstein	1997-03-24	0.25	Sales	Oxford	Oxford	United Kingdom
8	Hall	1997-08-20	0.25	Sales	Oxford	Oxford	United Kingdom
9	Olsen	1998-03-30	0.20	Sales	Oxford	Oxford	United Kingdom
10	Cambrault	1998-12-09	0.20	Sales	Oxford	Oxford	United Kingdom
11	Tuvault	1999-11-23	0.15	Sales	Oxford	Oxford	United Kingdom
12	King	1996-01-30	0.35	Sales	Oxford	Oxford	United Kingdom
13	Sully	1996-03-04	0.35	Sales	Oxford	Oxford	United Kingdom
14	McEwen	1996-08-01	0.35	Sales	Oxford	Oxford	United Kingdom
15	Smith	1997-03-10	0.30	Sales	Oxford	Oxford	United Kingdom
16	Doran	1997-12-15	0.30	Sales	Oxford	Oxford	United Kingdom
17	Sewall	1998-11-03	0.25	Sales	Oxford	Oxford	United Kingdom
18	Vishney	1997-11-11	0.25	Sales	Oxford	Oxford	United Kingdom
19	Greene	1999-03-19	0.15	Sales	Oxford	Oxford	United Kingdom
20	Marvins	2000-01-24	0.10	Sales	Oxford	Oxford	United Kingdom
21	Lee	2000-02-23	0.10	Sales	Oxford	Oxford	United Kingdom
22	Ande	2000-03-24	0.10	Sales	Oxford	Oxford	United Kingdom
23	Banda	2000-04-21	0.10	Sales	Oxford	Oxford	United Kingdom
24	Ozer	1997-03-11	0.25	Sales	Oxford	Oxford	United Kingdom

PENUTUP

1. Kesimpulan

Pada praktikum ini telah mempelajari DML atau Data Manipulation Language seperti SELECT, INSERT, UPDATE, DELETE dan juga Cara membuat Table, Join Table serta Constraint pada Table selain itu juga mempraktekan DML dengan Database lepkom dan menampilkan data sesuai dengan soal yang diminta.

2. SaranPenulisan

